Lending Club - Classification of Loan Status

Semih Barutcu

2/13/2021

Introduction

- The aim of this project is classifying loan status of accepted credits by Lending Club (An American peer-to-peer lending company, headquartered in San Francisco, CA). Related dataset can be found on Kaggle. https://www.kaggle.com/wordsforthewise/lending-club.
- The rejected credits include less features than the accepted credits but the lending company may seek additional information from potential borrowers to evaluate their expected payment status.
- Furthermore, classifying helps to get a better estimate of *Return on Investment* for the current credits.

Outline

1. Introduction
2. Data Discovery and Visualization
3. Logistic Regression
4. k-Nearest Neighbors
5. Decision Tree (C5.0 vs C5.0 Boosted)
6. Random Forest
7. Model Evaluation
8. Conclusion

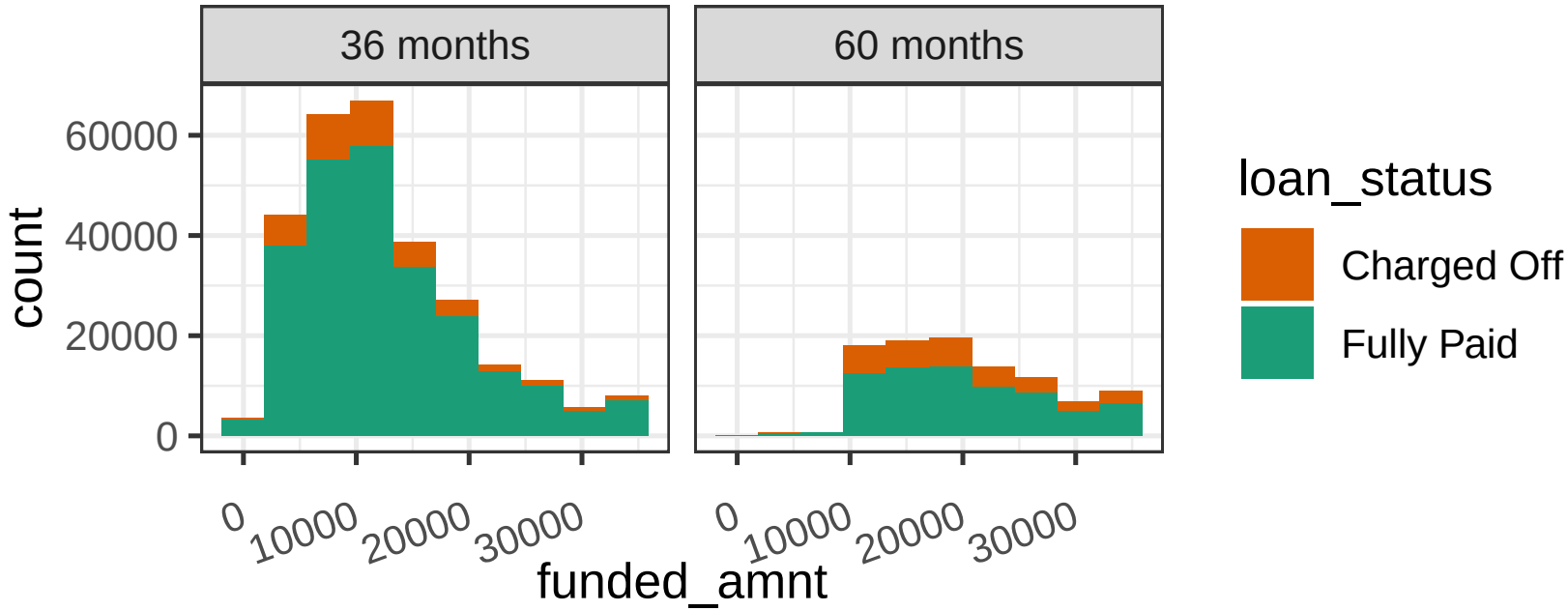
Data Insights

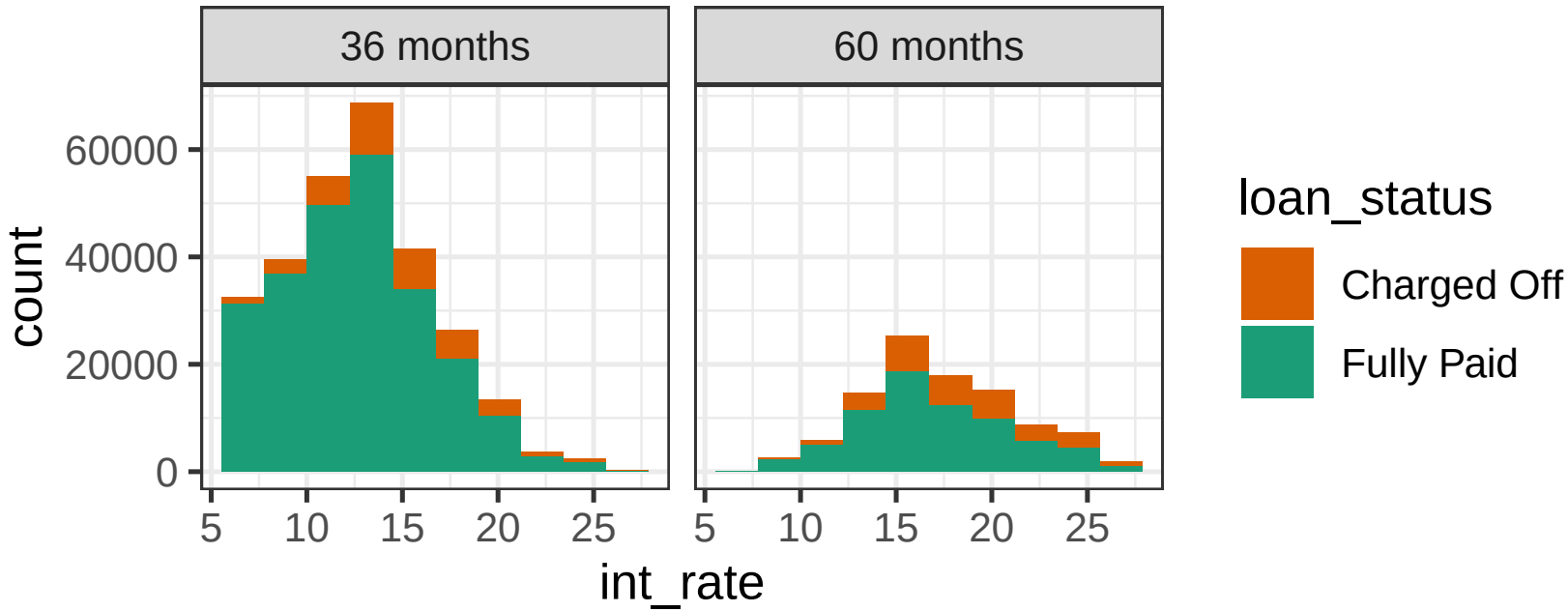
Logistic Regression, kNN, C5.0 and Random Forest algorithms are used to classify binary values of loan payment statuses which are charged-off (bad credits - default) and fully-paid (good credits).

This study is focused on a time range between 2012 and 2014. There are two different loan terms, 36 months and 60 months.

```
##
##                 36 months  60 months        Sum
##   Charged Off 0.09754597 0.07501069 0.17255666
##   Fully Paid  0.64271039 0.18473296 0.82744334
##   Sum         0.74025635 0.25974365 1.00000000
```

Visualization of the Data





Highlighted Features

- funded_amnt: The funded amount for the credit
- int_rate: The assigned int_rate for the credit
- annual_inc: The annual income of the borrower
- dti: The debt-to-income ratio of the borrower
- fico: The FICO credit score of the borrower
- grade: 7 different customer categories
- tot_cur_bal: The total current balance of the borrower

Preparing and Splitting the Data

- The data is imbalanced. Original proportions are 82.7% of fully-paid and 17.3% of charged-off. Good credits observations are under-sampled accordinglt total number of bad credits and a balanced dataset is obtained which has 50% of the each categories. This way, the resulting balanced dataset would provide a better learning process for any model.
- The balanced data has 130994 observations and 25 independent variables with the response variable loan_status. I used 75 to 25 percent split for training and test datasets. All of these models are tuned over a validation set sampled within training set without replacement.

Logistic Regression

Logistic Regression model regularized the learning well which gave nearly 85% of accuracies for train and test datasets. The table below and the tables you will see on the next pages show proportions of predictions. The false positives and false negatives have only 0.5% difference.

Accuracy = 84.92%

Cross Table	Predicted Charged Off	Predicted Fully Paid
Real Charged Off	0.426	0.073
Real Fully Paid	0.078	0.423

k-Nearest Neighbors

After a grid search for k value, k value is determined as 20 which gives more balanced results and mitigate over-fitting.

Total accuracy is significantly lower than Logistic regression results. The false positives and false negatives have 1.4% difference.

Accuracy = 79.77%

Cross Table	Predicted Charged Off	Predicted Fully Paid
Real Charged Off	0.405	0.094
Real Fully Paid	0.108	0.393

C5.0

C5.0 is a decision tree algorithm and the model gives 87.02% accuracy for the training data. As you can see, the gap between false positives and false negatives increased to 5.1%.

Accuracy = 85.14%

Cross Table	Predicted Charged Off	Predicted Fully Paid
Real Charged Off	0.450	0.049
Real Fully Paid	0.100	0.401

C5.0 Boosted

Boosted C5.0 includes 30 different trees and the model gives 90.73% accuracy for the training data although it does not improve test results. Because of increased over-fitting, it would be a better choice to use the single tree model.

Accuracy = 85.39%

Cross Table	Predicted Charged Off	Predicted Fully Paid
Real Charged Off	0.448	0.051
Real Fully Paid	0.095	0.406

Random Forest

Random Forest give slightly better accuracy levels than the Logistic Regression model and the C5.0 model. Also, the gap between false positives and false negatives increased to 6%.

These three algorithms have similar accuracies at the end but the decision criterion should consider costs of the false positives and false negatives. Assuming that the loss of a bad credit is higher than the profit of a good credit, Random Forest model seems to be a better choice of model.

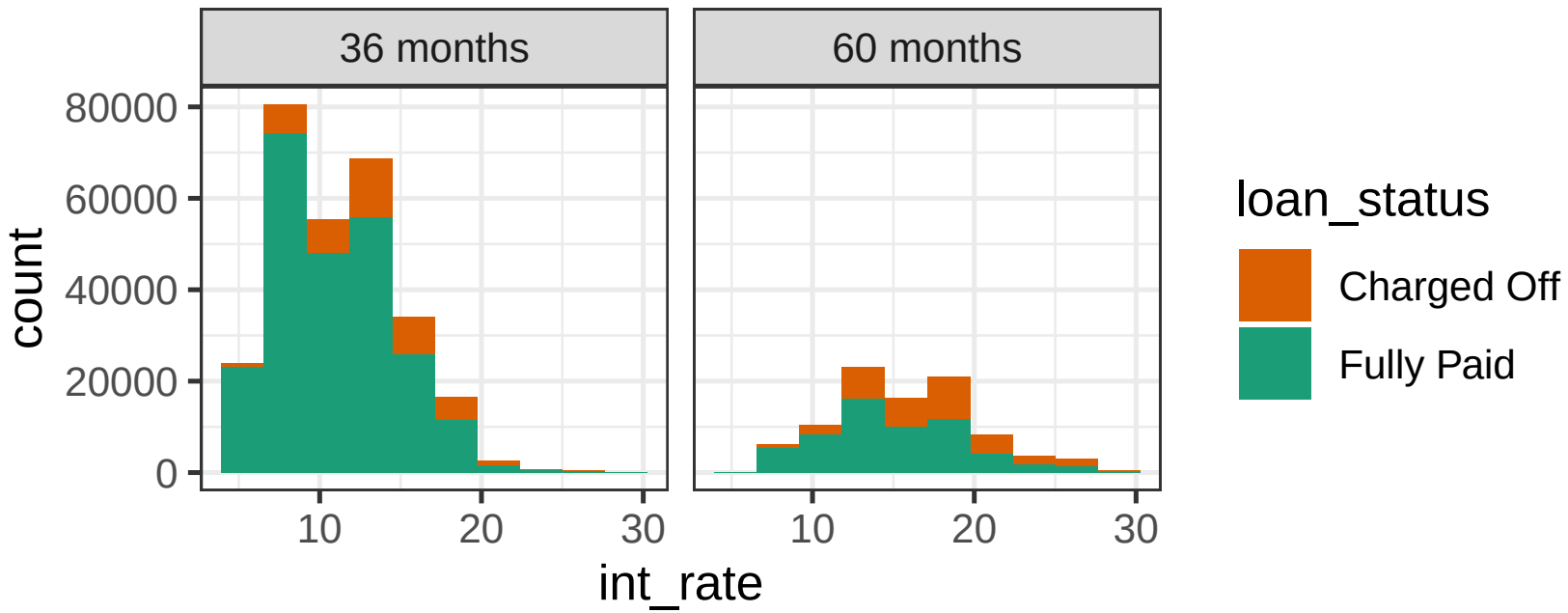
Accuracy = 85.41%

Cross Table	Predicted Charged Off	Predicted Fully Paid
Real Charged Off	0.456	0.043
Real Fully Paid	0.103	0.398

Model Evaluation: Lending Club 2015 Data

2015 Lending Club data includes higher proportions of charged-off category due to 60 months credits mostly have not resulted for fully-paid credits.

```
##
##                  36 months  60 months        Sum
##    Charged Off  0.11218689 0.08965951 0.20184639
##    Fully Paid   0.64145251 0.15670110 0.79815361
##    Sum          0.75363940 0.24636060 1.00000000
```



Performance Measures

- Sensitivity = True Positives / (True Positives + False Negatives)
- Specificity = True Negatives / (True Negatives + False Positives)
- Accuracy = (True Positives + True Negatives) / (True Positives + True Negatives + False Positives + False Negatives)

Testing All Models on 2015 Data

Performances of all models for an imbalanced data can be seen below. Logistic Regression has better accuracy than others for 2015. However, k-nearest neighbors has the highest specificity and Random forest has the highest sensitivity.

Models	Sensitivity	Specificity	Accuracy
Logistic Regression	0.906	0.882	0.887
k-Nearest Neighbors	0.711	0.920	0.811
C5.0	0.939	0.841	0.860
Random Forest	0.944	0.835	0.855

Conclusion

Trade-off of this study is between the cost of default and paid credits.

My last decision is to use Random forest, the method with the highest sensitivity. Detecting bad credits correctly 94.4% instead 90.6% is more valuable than getting higher total accuracy by 2.2%.

If the loss and profit values are known in advance, then an integer programming approach could be implemented to make the final model selection easier.